

Use Case # 1

Main success scenario (or basic flow) for Process Trips:

The user provides a CSV file containing train connection data. The system reads, validates, and stores the connections in memory. For each city, the system also tracks the incoming and outgoing connections.

Extensions (or alternative flows):

If the provided CSV file does not exist or is invalid, indicate error.

Use Case # 2

Main success scenario (or basic flow) for Filtering Trips:

The system asks the passenger to choose the method of filtering trips. The passenger agrees and is given the option to filter by Departure City, Arrival City, Departure Time, Arrival Time, Train Type, Days of Operation, First Class ticket rate (in euro) or Second Class ticket rate (in euro). The user can then modify the values for each search parameter.

Extensions (or alternative flows for Filtering Trips):

If no parameter has been chosen for filtering, indicate error.

If a DepartureCity or an ArrivalCity has not been chosen, indicate an error.

Use Case # 3

Main success scenario (or basic flow) for Search Trips:

After specifying the search parameters, the system displays all matching trips including direct and indirect connections (up to 2 stops). For each matching trip, the system also displays the trip duration, ticket price, and time to change (if applicable).

Extensions (or alternative flows) for Search Trips:

If there are no matching trips, provide a user message.

Use Case # 4

Main success scenario (or basic flow) for Sort Trips:

The system asks the passenger if they want to sort. The passenger agrees and is given the option between sorting by trip duration or by price. For the sorting by price, the system will allow the user to sort between the first class ticket cost and the second class ticket cost. An option will be given to allow the passenger to sort by ascending or descending order.

Extensions (or alternative flows) for Sort Trips:

If no identifier is entered or an invalid identifier is entered. Indicate an error or proceed as if the user has not chosen the sorting option. Display the results without implementing any sorting.

Use Case # 5

Main success scenario (or basic flow) for Book a Trip:

The System asks the client for which trip they want a reservation. The client chooses the trip shown by the system after they have filtered and/or sorted the connections. Once the trip has been chosen, the system asks the client for the name, age and id to create a reservation. The client enters the info and sends it to the system. The system then verifies that a client with the exact credentials already exists. If they do exist, the system will find the client object and use it to create a reservation and output a ticket to the user. If they do not exist, the system will create a new client object with the same info, which will then be used to create a reservation and output a ticket to the user.

Extensions (or alternative flows) for Book a Trip:

Suppose the client has chosen a trip but has not entered the passenger information. In that case, the system will either wait for the passenger information required to complete the reservation or reset to allow the user to choose a new trip/connection.

Use Case # 6

Main success scenario (or basic flow) for View a Trip:

The client can enter their last name and id to view all their current and past trips. Past trips should be placed in some 'history collection'.

Extensions (or alternative flows) for View a Trip:

If the client credentials are invalid, show an error message.

If the client has no trips, show an information message.